Selective Motion Estimation for Surveillance Videos

Muhammad Akram, Naeem Ramzan, and Ebroul Izquierdo

Electronic Engineering Department,
Queen Mary University of London,
United Kingdom
{muhammad.akram,naeem.ramzan,ebroul.izquierdo}@elec.qmul.ac.uk

Abstract. In this paper, we propose a novel approach to perform efficient motion estimation specific to surveillance videos. A real-time background subtractor is used to detect the presence of any motion activity in the sequence. Two approaches for selective motion estimation, GOP-by-GOP and Frame-by-Frame, are implemented. In the former, motion estimation is performed for the whole group of pictures (GOP) only when moving object is detected for any frame of the GOP. While for the latter approach; each frame is tested for the motion activity and consequently for selective motion estimation. Experimental evaluation shows that significant reduction in computational complexity can be achieved by applying the proposed strategy.

Keywords: Fast motion estimation, surveillance video, background subtraction, block matching algorithm.

1 Introduction

In surveillance applications, video captured by CCTV is usually encoded using conventional techniques, such as H.264/AVC. These techniques have been developed in view of conventional videos. With growing number of surveillance system deployments, there is a need to introduce surveillance centric coding techniques. Goal of this paper is to propose an efficient motion estimation approach specific to surveillance videos.

Motion is main source of temporal variations in videos. High compression efficiency is achieved through special treatment for motion based temporal variations. Motion estimation (ME) is a process that estimates spatial displacements of same pixels in neighboring frames. This spatial displacement is described through the concept of motion vector (MV). Almost all video coding standards deploy motion estimation modules to aid in removal of temporal redundancies. The process of ME divides frames into group of pixels known as block. Block matching algorithms (BMAs) are used to find out the best matched block from the reference frame within a fixed-sized search window. The location of the best matched blocked is described by MV. So, instead of encoding the texture of the block, only MV of the block is coded. While decoding video, motion vectors are used to replace the original blocks with its best matched block searched through motion estimation. Encoding complexity is

P. Daras and O. Mayora (Eds.): UCMedia 2009, LNICST 40, pp. 199–206, 2010.

dominated by the ME if full search is used as BMA. FS matches all possible displaced candidate blocks within search window to find a block with minimum block distortion measure (BDM).

Several fast BMAs have been introduced to beat FS in terms of computational complexity. These include new three step search (N3SS) [1], four-step search (4SS) [2], diamond search (DS) [3], kite-cross diamond search (KCDS) [4], and modified DS (MODS) [5], etc. In this paper, we propose a novel approach to reduce computational complexity for encoding surveillance videos. Proposed approach utilizes a real-time background subtractor (BGS) [6] to detect the presence of the motion activity in the sequence. In typical surveillance videos, scene remains static for a long period of time. Performing motion vector search for these frames is wastage of computing resources. Motion vector (MV) search is performed only for frames which have some motion activity identified by BGS.

In this paper, Section 2 introduces the implemented approach to perform efficient MV search for surveillance videos. Workflow of the proposed system is presented. Section 3 describes experimental results and presents a comparison of the proposed search with conventional search approach. Finally, Section 4 concludes this paper.

2 Selective Motion Estimation

The generic architecture of the implemented system is shown in Fig. 1. Surveillance video is presented to background subtraction and video encoding modules of the system. The real-time background subtractor detects motion activity present in the sequence. This information is passed onto the motion estimation module of the encoder. Motion estimation module utilizes the motion detection information to perform selective motion estimation. After motion compensated temporal filtering (MCTF) step, spatial transformation is performed to remove the spatial redundancies. Finally, entropy coding techniques are used to improve compression efficiency.

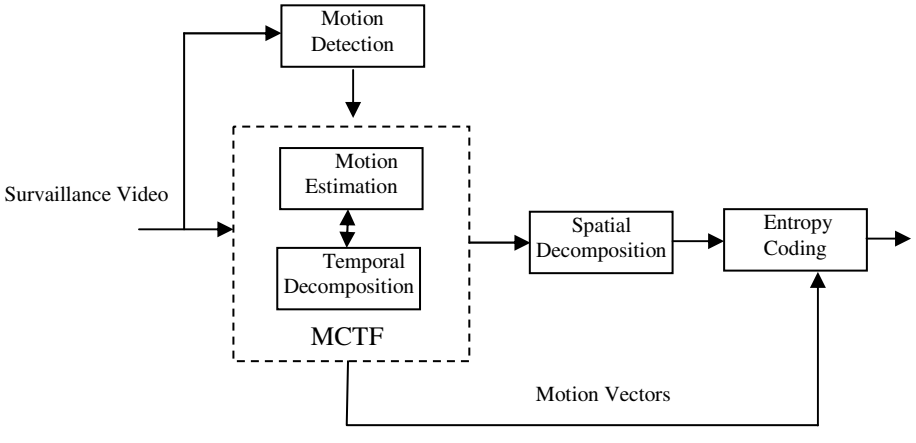

Fig. 1. Architecture of implemented system

2.1 Real-Time Background Subtraction

Motion detection module must be efficient in terms of processing power otherwise; the complexity of the motion estimation module shall be reduced at the cost of increased complexity for motion detection. A real-time video background subtraction module based on Gaussian mixture model [6] is used to detect motion activity present in the video. This method is able to deal robustly with light changes, bimodal background like swaying trees and introduction or removal of objects from the scene. Value of each pixel is matched against weighted Gaussians of mixture. Those pixels are declared as foreground whose value is not within 2.5 standard deviation of the background Gaussians. Foreground pixels are grouped into segmentation regions and bounded by rectangular boxes throughout the sequence. Output of BGS module for hall video is illustrated in Fig. 2. Pixels which are static for a number of frames are modeled as background; therefore they do not fall within the boundary of boxes. The presence of bounding box is an indication of motion activity present in the frame. This indication is used to perform selective motion estimation.

Fig. 2. BGS result for frame 122 of hall video

2.2 Selection Policy

As aforementioned, video captured from the CCTV camera is processed through a real-time background subtraction (BGS) module to detect the presence of motion activity in the sequence. Presence of motion activity for each frame of the sequence is marked and recorded. This information is utilized by the motion estimation module of the encoder to perform selective motion estimation. Two different selective motion estimation approaches, GOP-by-GOP and Frame-by-Frame, are implemented to improve the efficiency of motion estimation process in terms of saving processing power and processing time. In GOP-by-GOP approach, BGS information is analyzed for all the frames of every GOP. So, a single decision of performing selective motion estimation is made for each GOP. If BGS detects any moving object in any frame of the GOP then motion estimation is performed for the GOP otherwise motion vectors

for all the frames of the GOP are set to zero. Workflow of the proposed system is shown in Fig. 3. GOP-by-GOP selective motion estimation performs better when there is no motion activity for a large number of frames in the sequence. Its efficiency is lower when there is some pattern of activity present in the sequence not allowing to bypass the ME module. Also, this approach is dependent on GOP size set for encoding the sequence where smaller GOP size has better efficiency.

In Frame-by-Frame approach, BGS information for each frame of the sequence is analyzed. Decision of performing selective motion estimation is made for each frame. If any motion activity is present in a particular frame then motion estimation is performed otherwise motion vectors for the frame are set to zero. This approach improves processing efficiency by performing ME only for the frames where it requires and bypassing ME module for static frames. Thus, based on BGS analysis, no compromise is made for the frames which are important from surveillance standpoint and still complexity can be reduced by applying the proposed approach.

```
for (frame=1 to end of sequence)
  if (motion activity found)
    frameMotion [frame] = 1
  otherwise
    frameMotion [frame] = 0
end for

switch (motion estimation mode)
case GopByGop:
  for (each GOP of the sequence)
    for (first frame of GOP to GOP size)
      if ( frameMotion [frame] is 1)
        Perform Motion estimation for this GOP
        ME_performed = 1
        Break
    end for
    if ( ME_performed is not equal to 1)
        All the motion vectors are set to zero
  end for

case FrameByFrame:
  for (each frame of the sequence)
    if ( frameMotion [frame] is 1)
      Perform Motion estimation for this frame
   otherwise
        All the motion vectors are set to zero
  end for
```

Fig. 3. Algorithm for selective motion estimation

2.3 Video Coding

For actual encoding of the surveillance video, a wavelet-based scalable video codec – aceSVC [7] is employed. The scalable video coding (SVC) framework helps to improve utilization efficiency of available resources such as transmission bandwidth and storage, etc. Furthermore, SVC has potential for surveillance videos as in [8],[9]. Architecture of aceSVC features spatial, temporal, quality and combined scalability. Temporal scalability is achieved through repeated steps of motion compensated temporal filtering [10]. To achieve spatial scalability, each frame is decomposed using a 2D wavelet transform. Coefficients obtained through spatio-temporal decomposition are coded through the process of bit-plane coding [11] which provides basis for quality scalability.

3 Experimental Results

Performance evaluation of the proposed approach is carried out on three different surveillance sequences: “Dance”, “Hall” and “Street” with 500, 300 and 750 frames, respectively. All of these sequences have CIF (352x288) spatial resolution and frame rate of 30 Hz. Background in all the sequences is static throughout the length of the sequences. “Dance” sequence contains an animated person dancing with fast leg and arm motion. In “Hall”, two persons are walking in opposite directions in a corridor. In “Street”, with an outdoor street background, different animated objects move through the street.

While performing the experiment, sum of absolute difference (SAD) is used as block distortion measure (BDM). Block size is 16x16 while the search window size is 15 (+- 15 pel displacement is possible in vertical and horizontal directions). All the videos are compressed for 256 kbps bit-rate. Quarter pixel search approach is used for all the sequences. The true processing time is used to evaluate the performance of the proposed approach, while Y-PSNR is calculated to assess the image quality. As the surveillance video quality must be good enough for visual perception of the end user/observer; therefore, a subjective quality assessment test is also performed for the reconstructed sequences. The evaluation is performed using different GOP sizes. Each GOP contains at least one intra-coded frame. Thus, increasing the GOP size for the same sequence reduces the intra-coded frames in the whole compressed bit-stream. Consequently, higher GOP size has higher processing time.

All the tests are performed on machine with Intel Core(TM) 2CPU 6600@2.40GHz (2 CPU) processor and 2 GB RAM. First of all, BGS module has to be real-time to improve the efficiency of the proposed system. For this, Table 1 shows that the motion detection process is real-time where processing time for each surveillance sequence is given in seconds. BGS processes almost 30 frames in each second on the above described machine. Although BGS performance is real-time, still time consumed by BGS is included in overall encoding time for the evaluation of proposed selective motion estimation approach. In all the tables, PSNR results are in dB’s and time is measured in seconds.

Table 1. Real-time motion detection

Sequences	Total Frames	Time	Frames/Sec
Dance	500	17	29.41
Hall	300	10	30.00
Street	750	25	30.00

Experimental results for full search based motion estimation are summarized in Table 2. These results are used as reference to compare the proposed approach. Table 3 shows the results for GOP-by-GOP motion estimation. Different GOP sizes are selected to perform the experiment. With each GOP, MCTF is performed in such a way to produce maximum number of estimated frames. The processing time saving, compared to full motion estimation, achieved for GOP-by-GOP approach is shown in Table 4. Results show that the nature of the sequence has great influence on the efficiency of the proposed approach. One drawback with GOP-by-GOP motion estimation is that motion estimation is performed for all the frames of the GOP even if only one frame has the foreground object. Thus to refine and improve the performance, Frame-by-Frame selective motion estimation is implemented. Motion estimation is performed only for frames for which foreground object is detected. Table 5 and Table 6 show the experimental results for Frame-by-Frame approach. Results show significant improvement over GOP-by-GOP approach.

Table 2. Full motion estimation

Seq	Gop Size=8		Gop Size=16		Gop Size=32		Gop Size=64	
	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR
Dance	565	44.65	719	46.3	872	47.07	1027	47.37
Hall	438	35.53	556	37.62	678	38.59	819	39.19
Street	834	29.03	1085	32.13	1369	34.59	1692	36.19

Table 3. GOP-by-GOP selective motion estimation

Seq	Gop Size=8		Gop Size=16		Gop Size=32		Gop Size=64	
	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR
Dance	409	44.65	520	46.27	677	47.08	765	47.38
Hall	430	35.53	544	37.61	677	38.59	816	39.19
Street	810	29.04	1051	32.14	1334	34.59	1635	36.19

Table 4. Processing time saving for GOP-by-GOP selective motion estimation

Seq	Gop Size=8	Gop Size=16	Gop Size=32	Gop Size=64
Dance	27.61	27.68	22.36	25.51
Hall	1.83	2.16	0.15	0.37
Street	2.88	3.13	2.56	3.37

Table 5. Frame-by-Frame selective motion estimation

Seq	Gop Size=8		Gop Size=16		Gop Size=32		Gop Size=64	
	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR
Dance	320	44.61	398	46.24	481	47.04	600	47.35
Hall	430	35.53	549	37.61	665	38.58	804	39.18
Street	662	29.05	834	32.14	1053	34.61	1351	36.22

Table 6. Processing time saving for Frame-by-Frame selective motion estimation

Seq	Gop Size=8	Gop Size=16	Gop Size=32	Gop Size=64
Dance	43.36	44.64	44.84	41.58
Hall	1.83	1.26	1.92	1.83
Street	20.62	23.13	23.08	20.15

(a) (b)

Fig. 4. Visual comparison Hall frame 225 (a) Full ME (b) Frame-by-Frame ME

For assessing user perception based on visual quality, subjective quality evaluation based on the double stimulus impairment scale [12] method is performed as in Fig. 4. Different users participated in this test. Videos from full motion estimation, GOP-by-GOP motion estimation and Frame-by-Frame motion estimation were organized in random. User had to assign any number from 1 to 5 after watching the videos.

Table 7. Subjective quality result

Seq	Full ME	GOP-by-GOP	Frame-by-Frame
Dance	2.57	2.71	2.71
Hall	4.39	4.25	4.25
Street	2.82	2.68	2.53

Table 7 shows the results for visual evaluation of the sequences. These are average numbers where 5 is the maximum number representing the best quality. Results show that applying the proposed approach has no much effect on the visual perception of

the video which is important from the surveillance standpoint. This shows that the processing efficiency for the proposed approach is improved without compromising on visual quality of the surveillance videos.

4 Conclusion

In this paper, we have presented a novel technique to perform fast motion estimation specific to surveillance applications using the information of a real-time video background subtraction. Selective motion estimation is performed for GOP-by-GOP and Frame-by-Frame approaches. Performance of the implemented selective motion estimation approach is compared against motion estimation performed for all the frames. A high relative saving in processing time is achieved by using the proposed technique. Results obtained through experimental evaluation show that processing speed can be improved significantly by using the proposed approach while maintaining the surveillance sensitive information.

References

1. Li, R., Zeng, B., Liou, M.L.: A New Three-Step Search Algorithm for Block Motion Estimation. IEEE Trans. Circuit Syst. Video Technol. 4, 438–442 (1994)
2. Po, L.M., Ma, W.C.: A Novel Four Step Search Algorithm for Fast Block Motion Estimation. IEEE Trans. Circuit Syst. Video Technol. 6, 313–317 (1996)
3. Zhu, S., Ma, K.K.: A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation. IEEE Trans. Image Processing. 9, 287–290 (2000)
4. Lam, C.W., Po, L.M., Cheung, C.H.: A Novel Kite-Cross-Diamond Search Algorithm for Fast Block Matching Motion Estimation. In: IEEE ISCAS, vol. 3, pp. 729–732 (2004)
5. Yi, X., Ling, N.: Rapid Block-Matching Motion Estimation Using Modified Diamond Search. In: IEEE ISCAS, vol. 6, May 2005, pp. 5489–5492 (2005)
6. Stauffer, C., Grimson, W.E.L.: Learning Patterns of Activity Using Real Time Tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence 22, 747–757 (2000)
7. Mrak, M., Sprljan, N., Zgaljic, T., Ramzan, N., Wan, S., Izquierdo, E.: Performance Evidence of Software Proposal for Wavelet Video Coding Exploration Group. Technical Report, ISO/IEC JTC1/SC29/WG11/MPEG2006/ 13146 (2006)
8. Zgaljic, T., Ramzan, N., Akram, M., Izquierdo, E., Caballero, R., Finn, A., Wang, H., Xiong, Z.: Surveillance Centric Coding. In: 5th International Conference on Visual Information Engineering (VIE 2008), July 2008, pp. 835–839 (2008)
9. Akram, M., Ramzan, N., Izquierdo, E.: Event Based Video Coding Architecture. In: 5th International Conference on Visual Information Engineering (VIE 2008), July 2008, pp. 807–812 (2008)
10. Mrak, M., Izquierdo, E.: Spatially Adaptive Wavelet Transform for Video Coding with Multi-Scale Motion Compensation. In: IEEE International Conference on Image Processing, September 2007, vol. 2, pp. 317–3320 (2007)
11. Zgaljic, T., Sprljan, N., Izquierdo, E.: Bit-Stream Allocation Methods for Scalable Video Coding Supporting Wireless Communications. Signal Processing: Image Communications 22, 298–316 (2007)
12. Recommendation ITU-T BT 500.10: Methodology for the Subjective Assessment of the Quality of Televisions Pictures (2000)